**Problem Name:** Four sum II

**Topics:**

**Companies:**

**Level:** Easy

**Language:** C++

**Problem Statement**: Given four integer arrays nums1, nums2, nums3, and nums4 all of length n, return the number of tuples (i, j, k, l) such that:

* 0 <= i, j, k, l < n
* nums1[i] + nums2[j] + nums3[k] + nums4[l] == 0

**Input Format:**

First line of the input contain integer n (size of list)

Second line contain n space separated integer list values.

Last line contain integer value pos representing value of node to delete.

Ex:

5

1 2 3 4 5

1

**Output Format:** Print linked list after removing node having value pos

**Constraints:**

**Examples:**

**Brute force Solution:**

**Explanation:**  iterate over all the array & form all the **possible tuples** & see how many **tuples** are having **sum zero**

**Code:**

**Time Complexity**: O(n^4)

**Space Complexity: O(1)**

**Optimized Solution:**

**Explanation:**

* Here we are storing all possible addition of the first 2 vectors in a hashmap.
* Then we’re adding all combinations from the last 2 vectors and searched for the negative value from the map, it took O(1) time.
* If the value is present that means the addition of all 4 will be 0, so we increase the count.

**Code:**

**Time Complexity**: O(n^2)

**Space Complexity:** O(n)